A topology table 32.x or 34.x may for example comprise information concerning all neighbors within a certain semantical distance, or all neighbors that can be reached without having to pass more than k pointers (with k < 4, for example). It may also comprise a completely arbitrary collection of neighbors. A schematic representation of a topology table ~~32.2~~ 32.3 is shown in Figure 3B. In this example, the local neighborhood m=1 consists of all semantic units and pointers of segment 31, the local neighborhood m=2 consist of semantic units 30.2, 30.6, and pointers 33.1 and 33.3, while the local neighborhood m=3 consists of semantic units 30.2, 30.6, 30.7, and pointers 33.1, 33.3, and 33.8.

GM-B. 2/28/05

It is also possible to keep information derived or inherited from other semantic units in the topology table. Many other approaches for the definition of a topology table's contents are conceivable: all neighbors that can be reached via one particular type of connection object, for example, can be listed in a topology table; all semantic units that have the same string of letters, e.g. the same word stem, could be listed in the topology table; and so forth.

One can logically subdivide the knowledge database into segments. There can be an overlap between various segments. The topology and segments can be independent from each other.

For practical purposes the state of a node or pointer is often only a function of its local neighborhood. For example, when dealing with semantic networks a semantic unit is often connected to attributes, and the values of these attributes determine the state of the semantic unit.

After making all of the above definitions we are now in the position to define a self-organizing fractal semantic network, the fundamental model that we use in the present context.

A (locally) **self-organizing fractal semantic network** is a hierarchical, topological, higher-order, fractal, (locally) self-organizing semantic network.

In the following sections a set of basic building blocks and basic modules are defined, which are used to construct self-organizing fractal semantic networks.

As specified above, nodes and pointers of a network are called semantic units (cf. Figures 2A and 2B). All semantic units 100 are subdivided into concepts and instances. We further subdivide nodes into information units 101, attribute units 102, and module units or Janus Objects 103 and connection units or objects 104. Information units are general elements that can represent concepts or instances, and they are identified by specific

names. Attribute units 102 are identified by specific names and values, which can be set, retrieved, or computed.

Au.B. 2/28/05

All pointers of the network (connection objects 104) are either scaling (hierarchical connections 105) or non-scaling ~~sealing~~ (non-hierarchical connections 106). Standard inheritance principles are defined across all scaling pointers, making use of the network's topology or neighborhood concept. Pointers are further subdivided into comparison units 107, 109, interaction units 108, 110, description units 111, role units 112, and controller units 113. Non-scaling comparison units 109 allow us to describe the degree of similarity or dissimilarity of two semantic units, while scaling comparison units 107 allow us to describe how close one semantic unit comes to being an instance of another semantic unit, or how close one semantic unit comes to being a special case of another semantic unit. Non-scaling interaction units 110 allow us to describe the type of interaction of two semantic units, while scaling interaction units 108 allow us to describe the role one semantic unit plays as part of another semantic unit. Description units connect semantic units to their attribute units, which describe the semantic units in more detail. Role units describe the role one semantic unit plays with respect to another semantic unit. Finally, controller units connect semantic units to their Janus Objects 103, which in turn control and act upon the semantic units' local neighborhoods.

Au. B. 2/28/05

Figure 6 shows how the basic building blocks are used to construct a network. Note that each building block 61 labeled "Semantic Unit" can be replaced with any basic building block. Information units do not appear in this Figure 6, as there is no restriction on their use. In practice most of the building blocks 61 labeled "Semantic Unit" are information units.

A pointer 42.1 from a semantic unit 50.1 (herein referred to as object in order to be able to make a clear distinction between semantic units of the input network 18 and semantic units of the knowledge database 11) of the input network 18 to a semantic unit 40.3 in the knowledge database ~~18~~ 11 may carry a confidence value Cx. These pointers 42.x (cf. Figures 4A-4F and Figure 5) are herein referred to as "classification connections". If such a classification connection 42.x carries a confidence value Cx it corresponds to the classification probability, i.e. the probability that the object 50.1 from the input network 18 has been correctly matched with a semantic unit 40.3 in the knowledge database 11.

As described above, the knowledge database 11 used in the present context is a fractal semantic network of semantic units with local pervasive intelligence.

In accordance with the definition above, a network is called a fractal semantic network, if the following four conditions are satisfied:

attribute "mobile" now carries over through a classification connection 42.x to respective object 50.x in the input network 18. In doing so, the input network 18 is enriched with knowledge from the knowledge database 11.

Similarly, values of attributes may get inherited across hierarchies, where values are usually taken from the semantic unit closest to the one in question, as they can be regarded as more similar than units further away in hierarchy. For certain scaling or hierarchical connections 105 (is-in-particular connections or hierarchical similarity connections 107) roles or other closely linked neighbors can also be inherited.

JM.N. 2/28/08

During this inheritance step, some or all of the objects 50.x and/or pointers 51.x in the input network 18 may inherit across some or all of their classification connections 42.x the Classification Jani 43.x that are attached to the counterpart elements 40.x and/or 47.x in the knowledge database 11.

Note that the optional inheritance step and the classification step may at least to some extent be carried out concurrently.

Continuing with our example, the Classification Janus 43.1 copied to the semantic unit 50.2 performs the neighborhood analysis as follows. First it examines the counterpart unit's topology table 42.1. The topology table 42.1 is illustrated in Figure